Path Metric-based Navigation Protocol for Wireless Sensor Networks

Yuh-Ren Tsai, Sz-Yi Yeh and Kai-Jie Yang
Institute of Communications Engineering, National Tsing Hua University
Email: yrtsai@ee.nthu.edu.tw

Abstract

Wireless sensor networks (WSNs) are generally deployed in a desired area to monitor environmental parameters or to detect some specific events. WSNs are also applied to the applications of safety navigation. According to the gathered environmental information, a sensor node can guide users to an exit along the safest path. In previous work, the studies on the safety navigation issue are all focused on the detection of individual events. The objective of this work is to propose a path metric-based navigation protocol for WSNs. Each node detects and measures the metric of a target event, and then search for the safest path based on the proposed protocol. The search for the safest path is based on the path metric, defined as the sum of the measured metrics of all nodes in a path, and the path with the minimum path metric is chosen as the safest path. Furthermore, the proposed protocol is base on distributed algorithms in order to minimize the system complexity and to avoid the need of a central controller.

1. Introduction

Sensor networks are generally deployed in a desired area to monitor environmental parameters or to detect some specific events. If the connectivity among sensor nodes is maintained by wireless communications, it is known as a wireless sensor network (WSN) [1]-[4]. Recently, WSN are applied to the application of safety navigation. According to the gathered environmental information, a sensor node can guide users to an exit along the safest path. For example, a WSN may be deployed in an area to detect the room temperature. When a fire accident occurs, some adjacent nodes may discover that the detected temperature rises to an abnormal value, and then report the fire message to a remote base station (BS). In addition to sending out the fire alarm, the WSN should guide all the personnel to escape from the danger.

In previous work, the studies on the safety navigation issue are all focused on the detection of individual events [5]-[7]. Each sensor detects the event and measures the results in its sensor coverage. If a danger event occurs and is detected by a sensor node, this node will forward a danger message to all neighbor nodes. Upon this danger message, all the neighbor nodes will find an alternate path bypassing the danger node to a safe exit. Hence, all the personnel in this area can be guided by the instruction of this WSN to a safe place. These research works do not consider the situations in the entire path to the exits. However, in some applications such as the detection of gas, smoke or radiation, the major concern is to distinguish the accumulated densities in different paths. Moreover, it is possible that multiple events have occurred in a sensor network, and the network must find the comparatively safest path among all the paths having detected the danger events. For example, a path with a medium smoke density in a long range is worse than a path which has a little higher smoke density in a short range and a zero density in all rest range.

The objective of this work is to propose a path metric-based navigation protocol for WSNs. Each node detects the possible events and measures the results individually in its sensor coverage. Then, according to the proposed protocol, the safest path guiding to one of the exits is found for each node. The search for the safest path is based on the path metric, defined as the sum of the measured metrics of all nodes in a path, and the path with the minimum path metric is chosen as the safest path. In order to minimize the system complexity and to avoid the need of a central controller, the proposed protocol is based on distributed algorithms.

The remainder of this paper is organized as follows. Section II introduces some basic assumptions and represents the proposed path metric-based navigation protocol. In Section III the simulation results are presented to verify the proposed protocol. Finally, conclusions are drawn in Section IV.

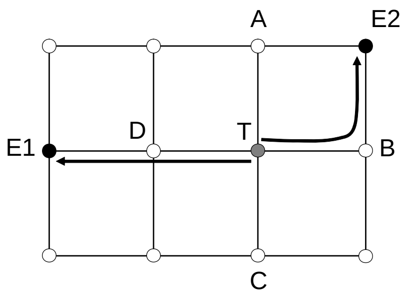
2. Path metric-based navigation protocol

It is assumed that the sensing area have a number of exits, and the entire area is well covered by a set of sensor nodes which can communicate with the neighbor nodes via a well defined wireless communication interface. Each sensor node is assumed to hold a unique identity (ID), and the IDs of the nodes deployed in exits are known by the system. Furthermore, each node maintains a database regarding the guiding path information, including the neighbor nodes' IDs and the

This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grants NSC 95-2752-E-007-003-PAE and NSC 94-2219-E-007-006.

path metric information corresponding to all exits, and a *History* register storing the identities of events that have been detected by this network.

Basically, there are possibly many paths from a node to an exit node, and all these paths must pass through one of the neighbor nodes. For all the possible paths passing through the same neighbor node, the minimum value of the path metrics corresponding to these paths is the major concern. Therefore, the path metric information concerns only the exit node, a neighbor node (referred to as an anchor node) and the corresponding minimum path metric. In other words, the path metric information can be represented as a triplet (*Anchor node ID, Path metric, Exit node ID*). From the viewpoint of a node, the number of paths destined to an exit node is equal to the number of neighbor nodes if finding the minimum metric path to an exit node is the major concern. If each node indicates the neighbor node which has the minimum path metric among all neighbor nodes, then the optimal path to a specific exit can be accurately found. For example, as shown in Figure 1, node T is the desired node and node E is the exit node. Node T has four neighbor nodes: node A, node B, node C and node D. Through each neighbor node, a corresponding sub-optimal route can be obtained, and thus node T maintains four pieces of path metric information for each exit. According to these four pieces of path metric information, the global optimal route from node T to an exit can be found by picking the neighbor node with the minimum path metric. In node T, the stored path metric information is shown in Table 1. Since there are two exits and four neighbor nodes for node T, the path metric information contains eight pieces of information.



E1 & E2: Exit node; T: Transmission node

Figure 1. An example of the sensor network architecture

Table 1. Stored Path Metric Information in Node T

Anchor node ID	*Path metric*	*Exit node ID*
Node A	6	Exit 1
Node B	4	Exit 1
Node C	4	Exit 1
Node D	2	Exit 1
Node A	2	Exit 2
Node B	2	Exit 2
Node C	4	Exit 2
Node D	4	Exit 2

2.1. Message formats

The path metric-based navigation protocol is divided into two phases: the initialization phase and the event-update phase. In the initialization phase, all sensor nodes search for the potential routes leading to the safe destinations. On the other hand, the event-update phase devotes to updating any new detected events, and then finds other alternate routes to an exit. Finally, after the event-update phase has been completed, the minimum metric path is shown to guide users safely going to the exit. Two messages are proposed for this protocol: the initialization message and the event-update message. The message formats are shown in Figure 2. The initialization message is a broadcast message with the corresponding fields defined as follows:

- "Type": indicates the type of this message, i.e. an initialization message or an event-update message;
- "Exit node ID": shows the exit node ID corresponding to this message;
- "Tx node ID": shows the ID of the node that is transmitting this message;
- "Path metric information": consists of all the stored path metric information corresponding to the exit node designated in the "Exit node ID" field. The path metric information is composed of the (*Anchor node ID, Path metric*) pairs;
- "Previous hop ID": shows the ID of the previous node that has transmitted this message to the transmitting node. This field is used to prevent the possibility of multiple message update.

Type	Exit node ID	Tx node ID	Path metric information		Previous hop ID
			Anchor node ID	Path metric	
			⋮	⋮	
			Anchor node ID	Path metric	

(a) The message format for the initialization message.

Type	Event identity			Update information		
	Event sequence number	Event occurring node ID	Tx node ID	Anchor node ID	Minimum Path metric	Exit node ID
				Anchor node ID	Updated Path metric	Exit node ID

(b) The message format for the event-update message.

Figure 2. The message formats

The event-update message is also a broadcast message with the corresponding fields defined as follows:

- "Type": indicates the type of this message;
- "Event identity": shows the event identity of the current event-update message, including the event sequence number, the event occurring node ID, and the transmission node ID, i.e. "Tx node ID";
- "Update information": shows the update

information corresponding to this event. The update information contains two triplets, including (*Anchor node ID, Minimum path metric, Exit node ID*) and (*Anchor node ID, Updated path metric, Exit node ID*), where the item "*Minimum path metric*" is the minimum metric corresponding to the specified exit, the item "*Updated path metric*" is the path metric that has been updated by this event, and the item "*Anchor node ID*" is the node ID corresponding to the minimum metric or the updated metric. If the minimum path metric is the same as the updated path metric, only one triplet is included.

2.2. Initialization phase

The protocol of the initialization phase is shown in *Algorithm 1*. In this phase, the potential routes leading to the safe destinations are found for all nodes. After the deployment of the sensor network has been accomplished, each destination node broadcasts an initialization message for all nodes to find all potential routes corresponding to this exit. Let E be a destination node deployed at an exit, the initialization message transmitted by E indicates "*Exit node ID*" = E, "*Tx node ID*" = E, "*Path metric information*" = (E, 0) and "*Previous node ID*" = E. After received the initialization message, a node neighbor with E stores the "*Path metric information*" in its memory, and then broadcasts a corresponding initialization message with "*Tx node ID*", "*Path metric information*" and "*Previous node ID*" fields being updated. Subsequently, all nodes will receive at least one initialization message corresponding to the exit node E. According to the received initialization messages, a node stores the minimum path metric corresponding to a pair of a neighbor node and an exit node. Thus the stored path metric information, as shown in TABLE I, can be constructed. Finally, after all nodes have received all the initialization messages, the initialization phase is successfully accomplished, and each node has a complete set of path metric information corresponding to itself and all exit nodes. Afterwards, each node indicates the moving direction of the path with the global minimum path metric to one of the exits. It is noted that the path metric is increased by 1 for each hop. Therefore, the path with the minimum path metric corresponds to the shortest path, and all users will be guided to the nearest exit.

Algorithm 1. The initialization phase protocol (find all potential routes to all exits)

```
Let E be a destination node deployed at an exit.
for all sensors S_i in the network do
    reset the path metric information, i.e. {(Anchor
     node ID, Path metric, Exit node ID)} = ϕ
E broadcasts an initialization message
if S_i receives an initialization message then
    if "Previous hop ID" is the same as S_i then
        do nothing
    else
        choose the "Path metric information" pair with
        the minimum path metric in the received
        message, excluding the one with "Anchor node
        ID = S_i"
        if no path metric corresponding to "Exit node
        ID" (E) and "Tx Node ID" then
            store the path metric information in memory:
            (Tx Node ID, Path metric + 1, E) ⇒
             (Anchor node ID, Path metric, Exit node ID)
            broadcast an initialization message
        else
            if the chosen "Path metric" + 1 is larger than
            the stored one with "Exit node ID" = E and
            "Tx Node ID" = stored "Anchor node ID" then
                do nothing
            else
                replace path metric information in memory:
                (Tx Node ID, Path metric+1, E) ⇒
                (Anchor node ID, Path metric, Exit node ID)
                broadcast an initialization message
                compare all path metrics and set the moving
                direction to the anchor node with the global
                minimum path metric
```

2.3. Event-update phase

The protocol of the event-update phase is shown in *Algorithm 2*. In this phase, the network responses to a target event, and updates the path metrics for all nodes. For a sensor node triggered by a target event, it translates the measured value into a detection metric, and then adds the detection metric to all path metrics stored in this node. Subsequently, the triggered node broadcasts event-update messages to diffuse this detection information to the entire sensor network. Since different exits form different sets of optimal routes, different event-update messages should be broadcast for different exit nodes. Let S_j be the node detecting a new event, the event-update message transmitted by S_j indicates "*Event identity*" = (sequence number, S_j) and "*Tx node ID*" = S_j. The message corresponds to a selected exit node, and the corresponding path metric information is used to fill the "*Update information*" field.

For any node received an event-update message, it verifies the "*Event identity*" to prevent any duplicate update. If this update is accepted, the corresponding path metric information will be updated. Then all the path metrics corresponding to the specified exit node are compared to find the minimum path metric for this exit. If the minimum path metric for this exit has been changed, this node will broadcast an event-update message to announce this updating. Afterwards, each node indicates the moving direction of the path with the global minimum path metric. It is noted that the value corresponding to a detection metric is much larger than 1; therefore, the selected path will be dominated by the target events, and all users will be guided to an exit through the safest route.

Algorithm 2. The event-update phase protocol (update the detection of the target events)

if a new event is detected in S_j **then**
  All the stored metrics are added by the detection metric of this new event
  broadcast an *event-update message* for each exit
if S_i receives an *event-update message* corresponding to an "*Exit node ID*" E **then**
  if the received "*Event identity*" has been stored in the *History* of S_i **then**
    do nothing
  else
    ignore the update information with "*Anchor node ID*" the same as S_i
    replace the stored "*Path metric*" corresponding to the "*Update information*" with the received "*Path metric*" + 1
    store the "*Event identity*" in the *History*
    if the minimum path metric corresponding to E has been changed **then**
      broadcast an e*vent-update message* to update the minimum path metric corresponding to E
      if another path metric corresponding to E is the same as the minimum one **then**
        broadcast another e*vent-update message* corresponding to E
      compare all path metrics and set the moving direction to the anchor node with the global minimum path metric
    else
      do nothing

It must be noted that the proposed navigation protocol is a fully distributed protocol, and no central controllers are required. Each node individually detects the occurrence of any target events, and independently responses the received messages. In addition, the proposed algorithms involve only very simple operations and comparison; therefore the proposed protocol is very suitable for the applications of WSNs.

3. Implementation and simulation results

In this section, we examine the path metric-base navigation protocol via simulation. As shown in Figure 3, the entire network consists of 15 sensor nodes equally distributed in the hallways of a building, and two of the nodes are deployed at the locations of the two exits, Exit 1 and Exit 2. According to *Algorithm 1*, the path metrics corresponding to different exits can be obtained. Figure 4 shows the path metrics stored in each node after the initialization phase has been accomplished. The path metrics in the brackets are shown in a clockwise order beginning with the upper direction branch. For example, as in Figure 3, the path metrics of node T are shown in the order (M_A, M_B, M_C, M_D), where M_i is the path metrics correspond to node *i*. Each node maintains different sets of path metrics for different exits. In Figure 4, the arrows show the possible guiding directions (with the minimum path metric in a node) from a node to a specific exit. Figure 4(a) and Figure 4(b) are the results corresponding to Exit 1 and Exit 2, respectively. It is noted that each node may have different guiding directions for different exits. Finally, the actual guiding direction of a node is determined by selecting the direction with a global minimum path metric. The results of the final guiding directions are shown in Figure 5.

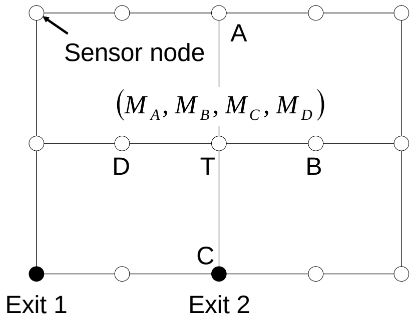
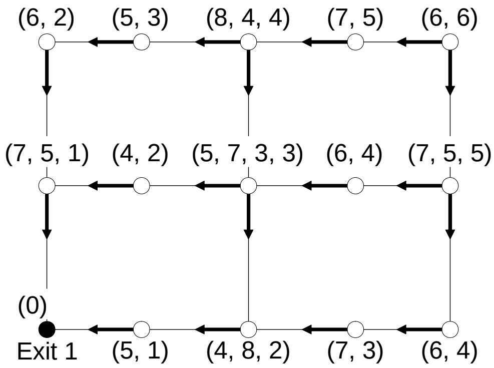
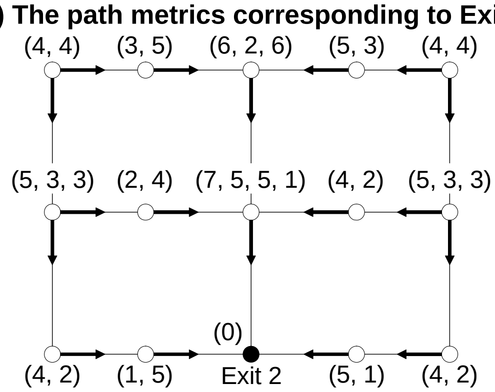


Figure 3. The sensor network topology used for simulation



(a) The path metrics corresponding to Exit 1



(b) The path metrics corresponding to Exit 2

Figure 4. The path metrics after the initialization phase has been accomplished

Considering Scenario 1 that a target event occurs at the location of node T with the detection metric being 100, all the path metrics stored in node T will be increased by an amount of 100. Subsequently, the event-update phase is initiated. Following *Algorithm 2*, the path metrics of all nodes will be updated. Figure 6 shows the path metrics stored in each node after the event-update phase has been accomplished. The metrics

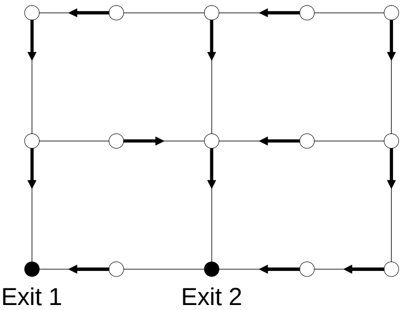
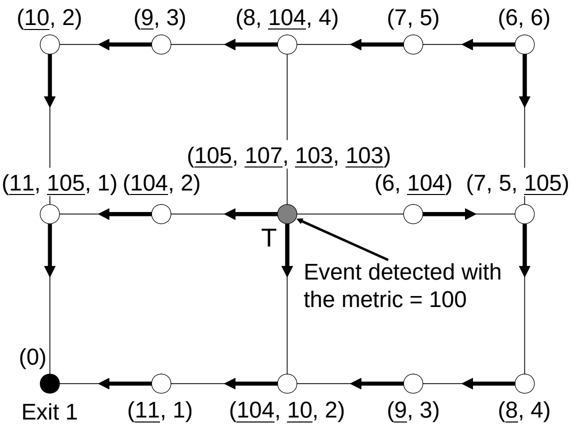
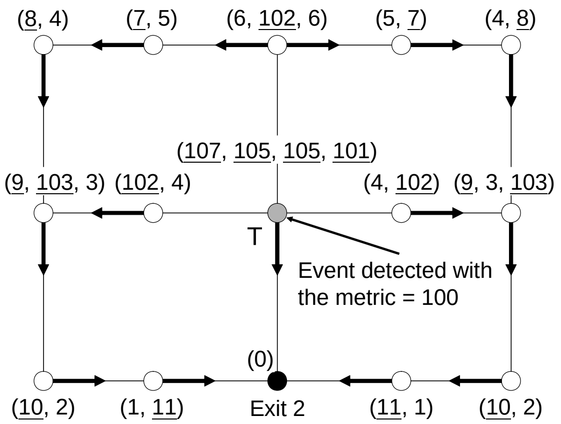


Figure 5. The guiding directions after the initialization phase has been accomplished

with underline are the ones that have been updated during the event-update phase. Furthermore, the guiding directions of some nodes are different from that shown in Figure 4. In other words, users will be guided to another path if the original path is influenced by the target event.



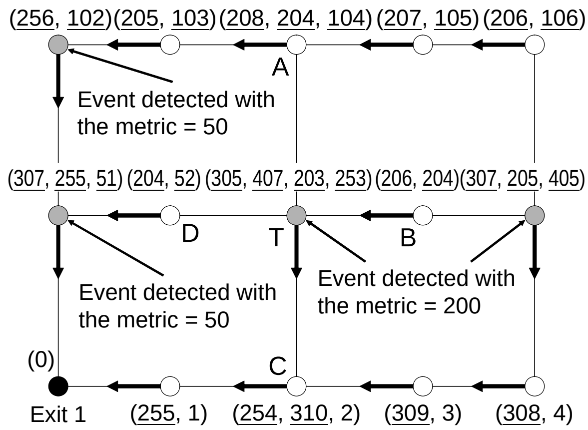
(a) The path metrics corresponding to Exit 1



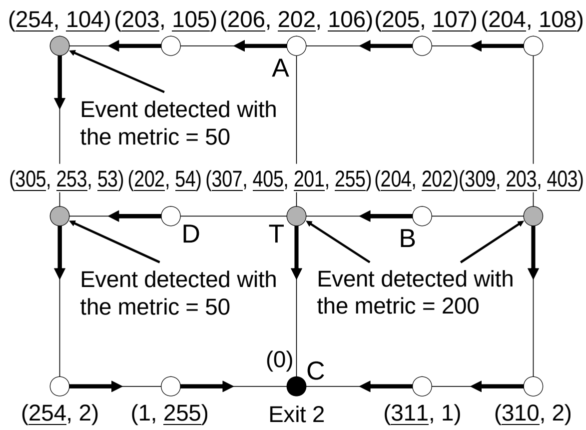
(b) The path metrics corresponding to Exit 2

Figure 6. The path metrics of Scenario 1 after the event-update phase is accomplished

Consider Scenario 2 that four events occur at the locations indicated in Figure 7. Two of the events are with the same detection metric 50, and the other two events are with the same detection metric 200. Figure 7 shows the path metrics stored in each node after the event-update phase has been accomplished. It was found that the sensor network will guide users to an exit along the path with the smallest path metric. For example in Figure 7 (b), the shortest path from node A to Exit 2 is passing through node T; however, this path does not have the minimum path metric. Therefore, the sensor network indicates a longer path, which has the smallest path metric, to Exit 2.



(a) The path metrics corresponding to Exit 1



(b) The path metrics corresponding to Exit 2

Figure 7. The path metrics of Scenario 2 after the event-update phase is accomplished

To further investigate the practicality of the proposed protocol, we apply our proposed scheme to a practical environment. As shown in Figure 8, there are a total of 72 sensor nodes deployed in an eight-floor building, which has two exits in the first floor. After the initialization phase has been accomplished, each node will guide users along the shortest path to one of the exits in the first floor. The guiding directions are shown in Figure 9(a). Assume that there are 7 sensor nodes, shown as gray nodes in Figure 9(b), having detected the target events. Then, after the event-update phase has been accomplished, each node will guide users along the path with the minimum path metric to one of the exits. To avoid the influence of target events, the sensor network may guide users to go to an upper floor and then to follow a safe path to an exit.

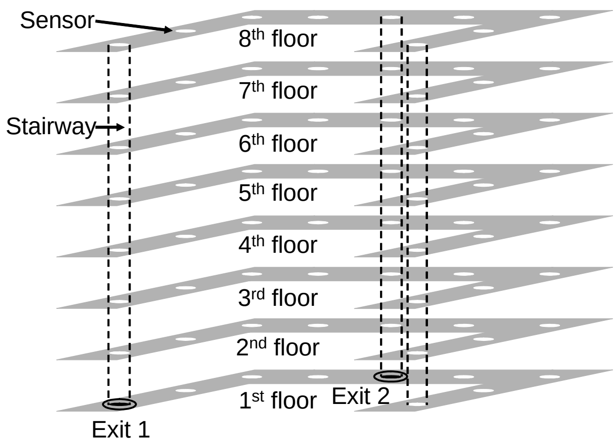
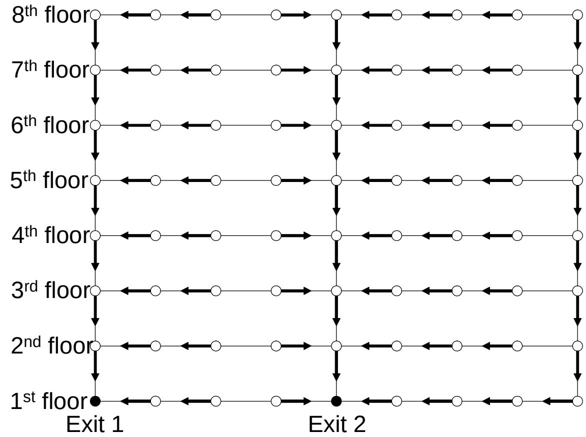
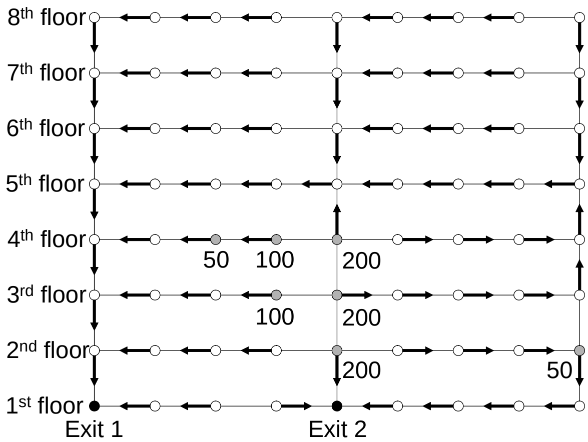


Figure 8. The deployment of a sensor network in an eight-floor building



(a) Guiding directions: the initialization phase



(b) Guiding directions: the event-update phase

Figure 9. The final guiding directions after the event-update phase has been accomplished

4. Conclusions

In this work, we have proposed a path metric-based navigation protocol for WSNs. The simulation results are provided to examine the feasibility of the proposed protocol. Moreover, our proposed protocol is applied to a practical environment to investigate the practicality. Our proposed protocol can be applied to the applications of detecting gas, smoke or radiation, and can distinguish the safest path to an exit according to the accumulated densities in different paths. Especially, our proposed protocol can find the comparatively safest path among all the paths having detected the danger events. In addition, the proposed algorithms are fully distributed algorithms involving only very simple operations and comparison. Therefore, the proposed protocol is very suitable for WSNs, and no central controllers are required.

References

[1] S. S. Iyengar, L. Prasad and H. Min, *Advances in Distributed Sensor Technology,* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[2] J. M. Kahn, R H. Katz and K S. J. Pister, “Mobile networking for smart dust,” in *Proc. ACM/IEEE International Conference on Mobile Computing and Networks,* pp. 271-278, 1999.

[3] D.Estrin and R. Govindan, “Next century challenges: scalable coordination in sensor networks,” in *Proc. ACM/IEEE International Conference on Mobile Computing and Networks,* Mobicom 1999, pp. 263-270, August 1999.

[4] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Communications of the ACM,* vol. 43, pp. 551-558, May 2000.

[5] Y. Hosokawa, N. Takahashi, and H. Taga, “A system architecture for seamless navigation,” in Proc. of *IEEE International Conference on Distributed Computing Systems Workshops*, pp. 498 – 504, 2004.

[6] A. Verma, H. Sawant, and J. Tan, “Selection and Navigation of Mobile Sensor Nodes Using a Sensor Network,” in Proc. of *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 41 – 50, 8-12 March 2005.

[7] Q. Li, and D. Rus, “Navigation Protocols in Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 3–35, August 2005.